

FPGA-based on-board Cubic Convolution Interpolation for Spaceborne Georeferencing

Dequan Liu^{1,2}, Guoqing Zhou^{1,2,3*}, Xiang Zhou^{1,3}, Chenyang Li^{2,4}, Fan Wang²

¹ School of Microelectronics, Tianjin University, Tianjin 300072 China. gzhou@glut.edu.cn (G.Z.); 1017232001@tju.edu.cn (D.L.); zqx0711@tju.edu.cn (X.Z.)

² The Center for Remote Sensing, Tianjin University, Tianjin, 300072 China. 2996868199@qq.com (W.F.)

³ GuangXi Key Laboratory for Spatial Information and Geomatics, Guilin University of Technology, Guilin, Guangxi, 541004, China

⁴ School of Marine Science and Technology Tianjin University, Tianjin, 300072 China. lichenyang_1008@tju.edu.cn (C.L.)

KEY WORDS: Cubic Convolution Interpolation, FPGA, Georeferencing, Remote Sensed Imagery, Real Time.

ABSTRACT:

This paper proposes a novel low-complexity implementation with one parameter two dimensions cubic convolution interpolation (2D-CCI) of image interpolation for spaceborne georeferencing. The experiment results demonstrate that (1) the proposed scheme can achieve better performance without increasing the computational complexity compared to the traditional CCI; (2) the number of multiplication and addition is reduced to 33.33% and 40% respectively compared with original CCI; (3) the occupied resources of LUTs, register, DSP and memory are reduced to 23.7%, 21.9%, 18.5% and 25%, respectively; (4) the speedup time of the proposed Implementation is about 25 times higher than that when using the PC-based implementation.

1. INTRODUCTION

Cubic convolution interpolation (CCI) is an originally process for the development for the reconstruction (i.e., interpolation, resampling) of Landsat digital image (Rifman, 1973, 1974; Robert, 1981; Park, 1982). The function has a good comprise between computational complexity and reconstruction. However, the traditional CCI consumes resource in hardware implementation. To solve this problem, this paper proposes an improved parallel pipelining two dimensional CCI (PP-2D-CCI) method based on field programmable date array (FPGA).

Geometric calibration is one of the most important steps for remotely sensed (RS) imagery (Zhou, 2018). With increasing demands in real-time or near real-time, such as military, disaster monitoring. The on-board implementation of RS processing has been investigated by many scientists and scholars (Zhou, 2017). To improve the georeferencing, this paper development the PP-2D-CCI scheme for real-time interpolation. The architecture of indirect georeferencing is described in Figure 1.

All the function modules are managed by the timing control module. Mathematical distortion module is used to choose a suitable mathematical distortion model, which describes the relationship between image coordinate and geodetic coordinate. The coordinate transformation module is implemented pixel-by-pixel through the image coordinate (X_{ref} , Y_{ref}), and the corresponding transformed coordinate (x , y) are calculated at the same clock. In general, the value of (x , y) coordinate will not be integer. Hence, a new coordinate must be estimated by an interpolation process. The paper focuses on the interpolation of the spaceborne georeferencing. For the details of other modules, it can be referenced to Zhou et al. (Zhou, 2010; Liu, 2019; Huang, 2018; Zhou, 2016).

The rest of work is organized as follow: Section 2 overviews the previously relevant work. Section 3 overviews of the cubic convolution interpolation and derives the proposed PP-CCI kernels. An FPGA-based implementation is presented in Section 4. The experiment and discussion are presented in Section 5. Finally, conclusions are given in the last section.

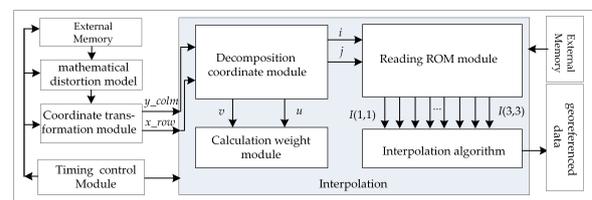


Figure 1. The architecture of georeferencing for RS imagery

2. RELEVANT WORK

To correct the distortion, it must essentially reposition pixels from their original location into a specified reference grid. RS Imagery georeferencing includes mathematical distortion module, coordinate transformation and resampling (interpolation) (Zhou, 2016). Interpolation is fundamental to digital image processing, especially in operations that require image resampling, such as scaling, registration, warping, and correction for geometric distortions (Schowengerdt, 2006). The convolution interpolation method for RS imagery georeferencing includes nearest-neighbor interpolation (Toutin, 2004), bilinear interpolation (Zhang, 2005), cubic convolution interpolation (Rifman, 1973, 1974; Robert, 1981; Park, 1982), and cubic B-spline interpolation (Arif, 2005). CCI function has been widely used to image interpolation and provided a good compromise between computational complexity and accuracy. Reichenbanch et al. (2001) proposed an improved 2D-CCI method for image reconstruction that is implemented the most general two-dimensional, non-separable, piecewise cubic interpolator with constraints for symmetry, continuity, and smoothness, and overcame the disadvantage of the traditional approach uses a separable convolution kernel. Meijering et al. (2018) established a link between classical interpolation and modern convolution-based interpolation and discussed the computational differences and given examples of other cubic interpolation schemes not previously studied in signal and image processing. Shi et al. (2003) presented 2D-3PCC and 2D-5PCC methods that were low-order polynomials with small spatial support and easy to implement and efficient to apply.

* Corresponding author

Zhou et al. (2016) proposed a novel edge directed CCI scheme which can be adapted to the varying edge structure of image. Hilal (2018) proposed a new resampling interpolation kernel which was dependent on five independent parameters that controlled its amplitude, angular frequency, standard deviation, and duration. Now, the CCI method are widely used in image processing, but the traditional CCI is still compute complex for hardware implementation.

FPGAs have been becoming the standard for on-board RS processing due to their smaller size, weight, and power consumption compared to other high-performance computing systems. However, the interpolation method for georeferencing on-board has not yet been reported worldwide so far. And conventional computing scheme are unsuitable for real time processing of image, since the computing power required to process large volumes of image data is enormous. To solve these problems, this paper presents an improved on-board cubic convolution interpolation based on FPGA for RS imagery georeferencing.

3. REVIEW AND OPTIMIZATION 2D CUBIC CONVOLUTION INTERPOLATION

3.1 Basic Concept of CCI Method

If $f(x)$ is a sampled function, $g(x)$ is the corresponding interpolation function. Mathematically speaking, the interpolation function in one dimensional (1D) is (Park, 1982):

$$g(x) = \sum_k c_k K\left(\frac{x-x_k}{h}\right) \quad (1)$$

where, c_k is the coefficient which is determined from the resampled data. h is the resampling increment. x_k represents the interpolation node, $K(s)$ signifies the interpolation kernel. Typically, $K(s)$ is identically zero for $|s| \geq 2$.

The basic function of the interpolation is that the value at the interpolation nodes or resample spots is equal to the value of the resampled data (Robert, 1981), i.e.,

$$f(x_k) = g(x_k) \quad (2)$$

CCI uses a separable piecewise cubic convolution (PCC) interpolation kernel with the constraint that it is continuous and has a continuous first derivative. These constraints result in the following kernel of Equation (3) (Rifman, 1973, 1974; Robert, 1981; Park, 1982).

$$K(s) = \begin{cases} 1 & s = 0 \\ (a+2)|s|^3 - (a+3)|s|^2 + 1 & 0 < |s| < 1 \\ a|s|^3 - 5a|s|^2 + 8a|s| - 4a & 1 < |s| < 2 \\ 0 & 1 = |s| \text{ or } |s| \geq 2 \end{cases} \quad (3)$$

The parameter a has a physical significance in Equation (3), it is the slope of $K(s)$ at $s=1$. The common value is $a=-1$ (Rifman, 1973, 1974), which duplicates the slope of the ideal interpolator at $s=1$ (also called standard cubic convolution), however, it was proven that (Robert, 1981) $a=-0.5$ is superior to $a=-1$ in that it provides an interpolation with better convergence properties. An intermediate choice, $a=-0.75$ (Park, 1982), is also sometimes advocated, this choice results from forcing Equation (3) to have second derivative continuity at $s=1$ (Hou, 1978). The various members of interpolation

family are illustrated in Figure 2 (time domain). Since $K(s)$ is an even function, only values $s \geq 0$ is shown in Figure 2.

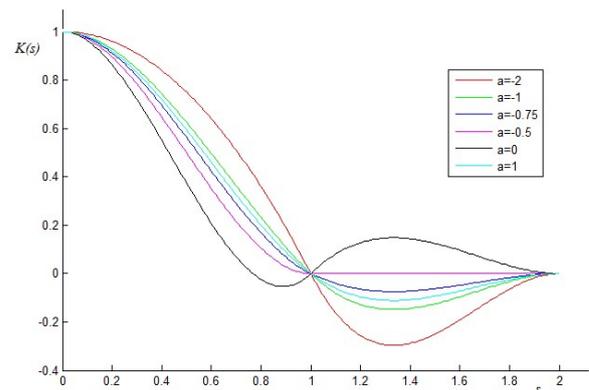


Figure 2. Six number of the family of $K(s)$.

Now suppose that x is any point at which the sampled data is to be interpolated, then x must be between two consecutive interpolation nodes which can be denoted $[x_j, x_{j+1}]$. Equation (3) is substituted into Equation (1), the CCI function is obtained.

$$g(x) = c_{j-1}K\left(\frac{x-x_{j-1}}{h}\right) + c_jK\left(\frac{x-x_j}{h}\right) + c_{j+1}K\left(\frac{x-x_{j+1}}{h}\right) + c_{j+2}K\left(\frac{x-x_{j+2}}{h}\right) \quad (4)$$

Since $s = (x - x_j) / h$ and $0 < s < 1$, Equation (4) reduces to (Shlien, 1979; Bailey, 2011):

$$g(x) = c_{j-1}K(s+1) + c_jK(s) + c_{j+1}K(s-1) + c_{j+2}K(s-2) \quad (5)$$

The resulting one-dimensional interpolation weights two samples on either side of the desired data point, giving a region of support of four samples. Extending a one-dimensional cubic convolution to two dimensional (2D) cubic convolution interpolation, which requires sixteen (4×4) spots near the resampled spot. Figure 3 displays the processing of the 16-2D-CCI.

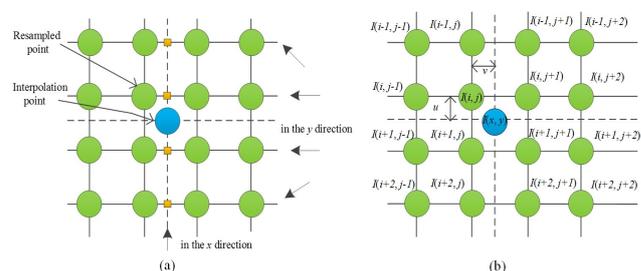


Figure 3. Cubic convolution interpolation. (a) Cubic Convolution interpolation one times in the y direction. (b) 2D Cubic convolution interpolation.

2D-CCI function as the following formula shows:

$$I(x,y) = I(i+v, j+u) = \sum_{row=-1}^2 \sum_{colm=-1}^2 f(i+row, j+colm) K(v-row) K(u-colm) \quad (6)$$

Equation 6 is rewritten as matrix form,

$$I(x, y) = [K(v+1) \ K(v) \ K(v-1) \ K(v-2)] \times \begin{bmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) & f(i+1, j+2) \\ f(i, j-1) & f(i, j) & f(i, j+1) & f(i, j+2) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) & f(i+1, j+2) \\ f(i+2, j-1) & f(i+2, j) & f(i+2, j+1) & f(i+2, j+2) \end{bmatrix} \quad (7)$$

$$\times [K(u+1) \ K(u) \ K(u-1) \ K(u-2)]^T = Kv \times I \times Ku$$

where, (x, y) is coordinate of the image after interpolation. i and j are the integer part of x and y , respectively; v and u are the fractional part of x and y , respectively. $Kv =$

$$\begin{bmatrix} K(v+1) & K(v) & K(v-1) & K(v-2) \end{bmatrix}, \quad Ku = \begin{bmatrix} K(u+1) & K(u) & K(u-1) & K(u-2) \end{bmatrix}^T \text{ and } I = \begin{bmatrix} I_{i-1, j-1} & I_{i-1, j} & I_{i-1, j+1} & I_{i-1, j+2} \\ I_{i, j-1} & I_{i, j} & I_{i, j+1} & I_{i, j+2} \\ I_{i+1, j-1} & I_{i+1, j} & I_{i+1, j+1} & I_{i+1, j+2} \\ I_{i+2, j-1} & I_{i+2, j} & I_{i+2, j+1} & I_{i+2, j+2} \end{bmatrix}$$

3.2 Optimization of 2D-CCI Function

3.2.1 Designing Memory Access: In order to improve the efficiency of memory access, a multi-array memory architecture is designed. Assuming the size of RS image is $m \times n$. As such $m \times n$ memory unites are designed in double data rate (DDR). The relationships between the address of DDR and (i, j) coordinate can be described as Equation (8).

$$addr_{ij} = (i-1) \times m + (j-1) \quad (8)$$

where, i and j are the row and column of the raw image, respectively. $addr_{ij}$ is the corresponding address of the memory.

3.2.2 Optimization the kernel function: Substituting the weight into equation (3) and simplifying by the power of the weight, Equations (9) thru (10) can be defined as (Robert, 1981).

$$K(s+1) = as^3 - 2as^2 + as \quad (9)$$

$$K(s) = (a+2)s^3 - (a+3)s^2 + 1 \quad (10)$$

$$K(s-1) = -(a+2)s^3 + (2a+3)s^2 - as \quad (11)$$

$$K(s-2) = -as^3 + as^2 \quad (12)$$

Substituting Equation (9) thru (12) into (7), Equation (13) can be obtained.

To implement the weights of 2D-CCI scheme based on Equation (13), according to the relationship between the weight formula and its coefficients, the nine weight formulas are divided into four levels. The first level includes $a+3$, $a+2$, av , u^2 , and v^2 . The second level includes u^3 , v^3 , av^2 , au^2 , $(a+2)v^2$, $(a+2)u^2$, $(a+3)v^2$, and $(a+3)u^2$. The third level includes $2av^2$, $(a+2)v^3$, $(a+2)u^3$, av^3 , au^3 , $(2a+3)v^2$, and $(2a+3)u^2$. The four level includes $av-2av^2+av^3$, $1-(a+3)v^2+(a+2)v^3$, $-av-(2a-3)v^2-(a+2)v^3$, $-av^3+av^2$, $au-2au^2+au^3$, $1-(a+3)u^2+(a+2)u^3$, $-au-(2a-3)u^2-(a+2)u^3$, and $-au^3+au^2$. The weight coefficients u and v have the same structure, and the optimization structure of v is

shown in Figure 4. Nine multipliers and nine adders are used to calculate the weight of v . Additionally, the values of $-au$, $-av$, $-2av^2$, $-2au^2$, av^3 , $-au^3$, $-(a+2)u^3$, and $-(a+2)v^3$ can be obtained by negating the corresponding sign bit of au , av , $2av^2$, $2au^2$, av^3 , au^3 , $(a+2)u^3$, and $(a+2)v^3$, respectively. Comparing with the original weight functions, thirteen multiplies and six adders are reduced, respectively.

$$I(x, y) = \begin{bmatrix} av - 2av^2 + av^3 \\ 1 - (a+3)v^2 + (a+2)v^3 \\ -av + (2a+3)v^2 - (a+2)v^3 \\ av^2 - av^3 \end{bmatrix} \times \begin{bmatrix} I(i-1, j-1) & I(i-1, j) & I(i-1, j+1) & I(i-1, j+2) \\ I(i, j-1) & I(i, j) & I(i, j+1) & I(i, j+2) \\ I(i+1, j-1) & I(i+1, j) & I(i+1, j+1) & I(i+1, j+2) \\ I(i+2, j-1) & I(i+2, j) & I(i+2, j+1) & I(i+2, j+2) \end{bmatrix} \times \begin{bmatrix} au - 2au^2 + au^3 \\ 1 - (a+3)u^2 + (a+2)u^3 \\ -au + (2a+3)u^2 - (a+2)u^3 \\ au^2 - au^3 \end{bmatrix} \quad (13)$$

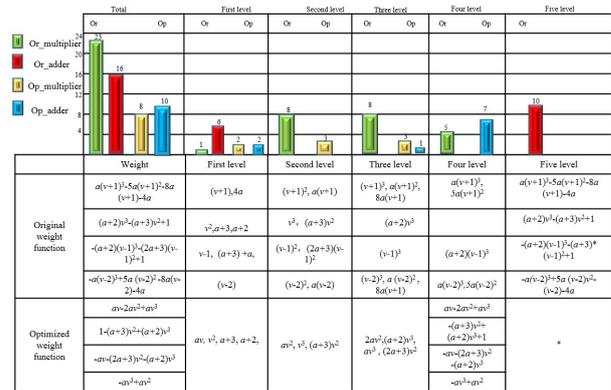


Figure 4. Optimization weight of v and u . Or: Original; Op: Optimized.

4. FPGA IMPLEMENTATION

To implementation Equation (13) on an PFPA, Equation (13) can be simplified into Equation (14).

$$I(x, y) = [Kv_{11}I_{11} + Kv_{12}I_{21} + Kv_{13}I_{31} + Kv_{14}I_{41}]Ku_{11} + [Kv_{11}I_{12} + Kv_{12}I_{22} + Kv_{13}I_{23} + Kv_{14}I_{42}]Ku_{21} + [Kv_{11}I_{13} + Kv_{12}I_{23} + Kv_{13}I_{33} + K(v-2)I_{34}]Ku_{31} \quad (14)$$

From Equation (14), the proposed architecture of PP-2D-CCI is described in Figure 5, which are consist of four modules: decomposition coefficient module, Reading ROM module, Calculation weight module, and cubic convolution interpolation module.

The function of decomposition coordinate module aims to generate the ROM address (the integer part of the coordinates x_{row} and y_{col} , respectively) and the weights (the fractional part of the coordinates x_{row} and y_{col} , respectively) that are needed to be interpolated. The reading ROM module is used to compute sixteen ROM address and read the corresponding value. The computation weight module parallel calculates the weight of the interpolation point in the horizontal and vertical directions according to the interpolation kernel. CCI module is

used to accomplish the piecewise cubic convolution interpolation. All the function modules are managed by the timing control module.

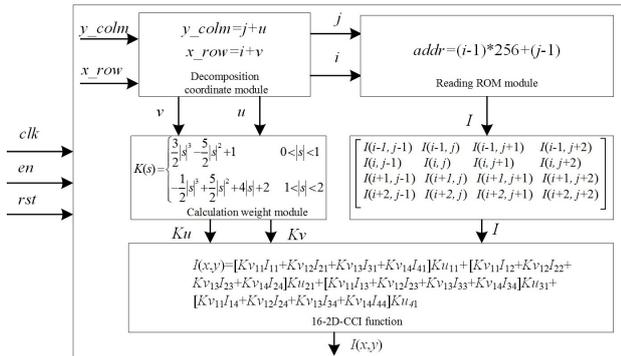


Figure 5. The FPGA-based system architecture

4.1 Parallel Decomposition of (x_row, y_colm) Coordinate

Generally, the coordinate is floating-point, but the ROM address is integer, in order to reduce the resource consumption of the FPGA, the strategy of mixed data type is adopted, and the fixed-to-float IP core is used to realize the transformation. The specific flowchart is shown in Figure 6. Figure 7 displays the parallel decomposition of coordinate, which includes six subtraction and two absolute operation are implemented through Xilinx’s IP cores. x_row and y_colm are parallel decomposed at the same clock.

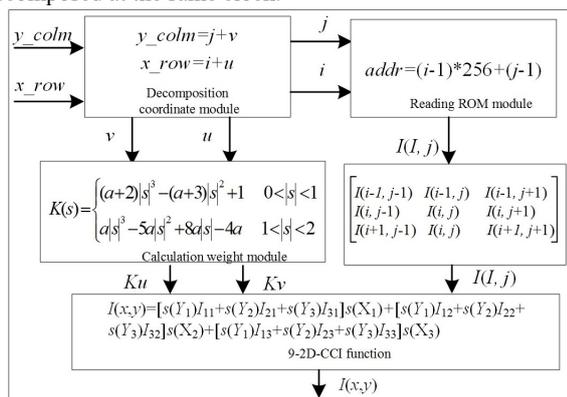


Figure 6. The flowchart of decomposition of coordinate.

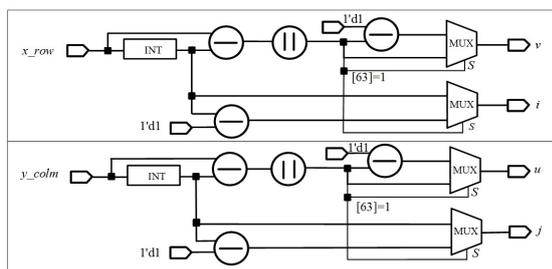


Figure 7. Parallel decomposition of the coordinate.

4.2 FPGA-Based Parallel Reading Memory

From Equation (8), An architecture of reading memory is designed, which includes implementation address and reading memory. The structure is illustrated in Figure 8. A strategy for converting multiplication operations into left-shift operations is adopted, which not only reduces the consumption of FPGA resources, but also raises the speed of operation. Eleven adders,

two subtractions, and three RTL left shifts are used to implement this system.

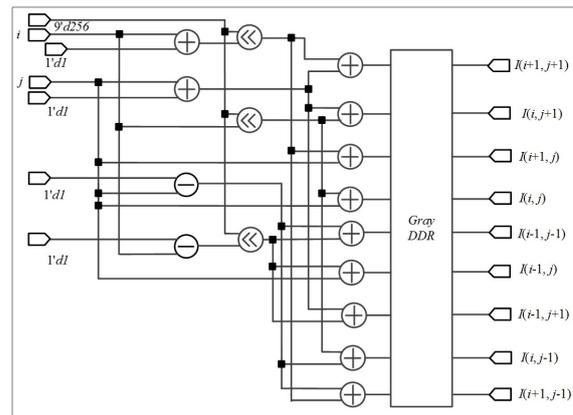


Figure 8. FPGA-based reading DDR memory

4.3 Parallel Calculation Coefficients

Figure 9 shows the architecture of parallel calculating the weights u and v, sixteen multiplies, eight adders and six add3 modules (containing twelve adders) are used. At the timing controlling, the results are output into the next module (interpolation module) at the same clock cycle.

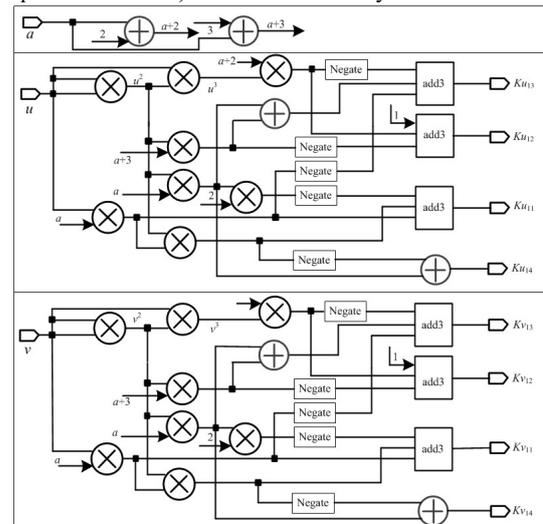


Figure 9. Parallel calculation weights

4.4 FPGA-Based Parallel Compute the Interpolation

After implementing the matrices of K_u , K_v and I , the value of interpolation can be computed from Equation (14). The parallel architecture was designed in Figure 10. Twenty multipliers, and fifteen adders are used in this architecture.

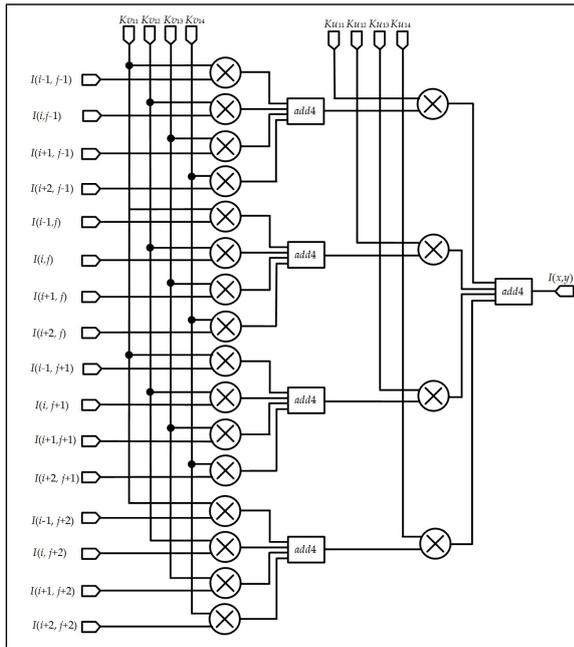


Figure 10. The architecture of PP-CCI.

5. EXPERIMENT

5.1 The Software and Hardware Environment

The proposed architecture is implemented on a custom-designed board which contains a Xilinx Artix-7 XC7VX980-tffg1930-1 FPGA. The selected FPGA has 612,000 Logic Cells, 1,500kb Block RAMS, 1,224,000 Flip-Flops and 3,600 DSP Slices. In addition, the design tool is Vivado 2014.2, the simulation tool is ModelSim SE-64 10.4, and hardware design language is Verilog HDL. To validate the proposed method, the georeferenced algorithm is also implement using MATLAB R2014a on PC with Windows 7(64 bit) operation system, which is equipped with an Inter(R) Core i7-4790 CPU @ 3.6 GHz and 8 Gb RAM.

5.2 Data

To validate the proposed system based on an FPGA, 256*256 pixels SPOT TM data is used to perform the interpolation for Georeferenced image. The data set is bldt_tm.img from the ENVI software, with its own control points file is bldt_tm.pts. The bldt_tm.img image with the UMT protection, the zone of 13° , the resolution of 30m, the band of three, the wavelength range of $0.63 \sim 0.69 \mu\text{m}$. Figure 11 displays the raw image.

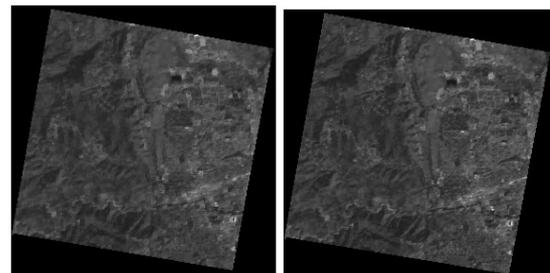


Figure 11. The raw image

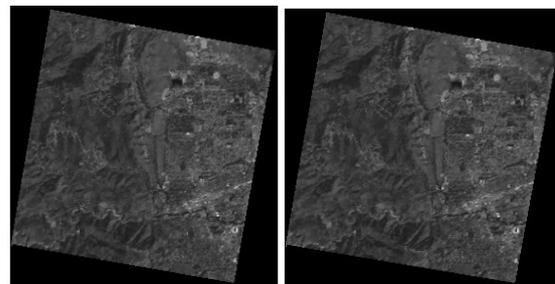
5.3 Processing Performance Analysis

5.3.1 Visual check: (1) Comparison with different parameter a of PP-2D-CCI method based on FPGA

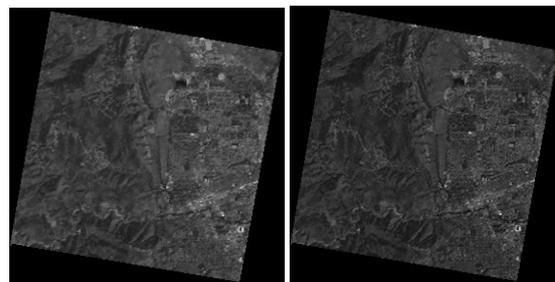
Various members of this family of interpolators corresponding to $a=1, 0, -0.5, -0.75, -1$, and -2 are compared with the same data, as shown in Figure 12. By visual inspection, when $a=1$, and 0 , the interpolated image texture information is lost more, and the interpolation effect is significantly reduced. When $a=-0.5, -0.75, -1$, and -2 , the interpolated image contains more texture information, and the interpolation effect is better than that of $a=1$, and 0 .



(a) (b)



(c) (d)



(e) (f)

Figure 12. The PP-2D-CCI for georeferenced imagery interpolation with different parameters a based on FPGA. (a) $a=1$. (b) $a=0$. (c) $a=-0.5$. (d) $a=-0.75$. (e) $a=-1$. (f) $a=-2$.

(2) Comparison with different interpolation method

To validate the accuracy of proposed interpolation method, nearest neighbour interpolation, bilinear interpolation, and CCI algorithms are executed based on MATLAB. The interpolation images of nearest neighbour interpolation, and bilinear interpolation are shown in Figure 13. Through the visual check Figure 13 and 12, it can be found that the result of CCI interpolation georeferenced image has the enough smoothness and visual effect is better than that of the bilinear interpolation and nearest neighbour interpolation. However, the CCI algorithm consumes more resources.

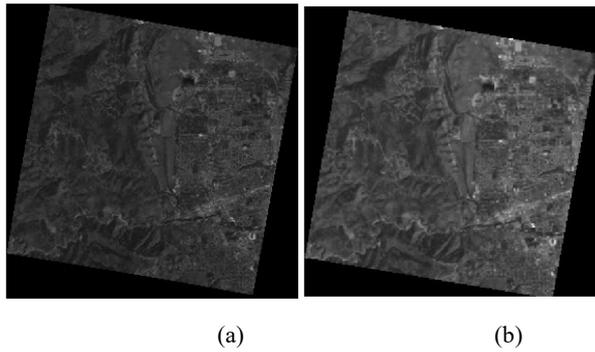


Figure 13. Comparison with different interpolation methods. (a) Nearest neighbour interpolation. (b) Bilinear interpolation.

5.3.2 Accuracy of gray value: To compare PP-2D-CCI scheme with the different a . The forty check points (CPs) are selected on the georeferenced image. The distribution of CPs is shown in Figure 14. Table 1 lists the comparison of interpolation accuracy. It can be found that the gray value of CPs obtained by PP-2D-CCI are approximately equal, respectively.

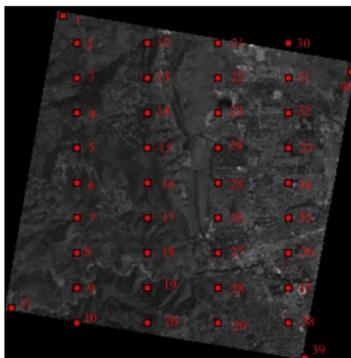


Figure 14. The distribution of the forty CPs labelled as read in the georeferenced image.

Table 1. The comparison of gray value of Cps.

serial number	(row, column)	$a=0$	$a=0.75$	$a=0.5$	$a=0$	$a=1$	serial number	(row, column)	$a=0$	$a=0.75$	$a=0.5$	$a=0$	$a=1$
1	(45,6)	46	47	48	48	49	21	(168,28)	61	60	59	59	58
2	(56,28)	43	43	43	43	42	22	(168,56)	59	58	58	58	57
3	(56,56)	38	37	37	37	37	23	(168,84)	60	60	60	60	59
4	(56,84)	29	30	30	30	31	24	(168,112)	55	55	55	55	55
5	(56,112)	35	36	36	36	36	25	(168,140)	53	52	52	52	52
6	(56,140)	36	35	35	35	35	26	(168,168)	67	65	64	64	63
7	(56,168)	48	48	47	47	47	27	(168,196)	39	39	39	39	39
8	(56,196)	61	61	61	61	61	28	(168,224)	50	48	48	48	46
9	(56,224)	35	36	36	37	38	29	(168,252)	42	41	40	40	39
10	(56,252)	0	0	0	0	0	30	(224,28)	0	0	0	0	0
11	(4,240)	53	56	57	57	58	31	(224,56)	66	67	67	67	68
12	(112,28)	34	36	36	37	37	32	(224,84)	50	50	50	50	50
13	(112,56)	29	30	30	30	31	33	(224,112)	40	42	43	43	44
14	(112,84)	29	29	29	29	29	34	(224,140)	77	76	75	75	74
15	(112,112)	53	51	50	49	48	35	(224,168)	85	84	84	84	84
16	(112,140)	43	44	44	44	44	36	(224,196)	69	66	65	64	63
17	(112,168)	46	47	47	47	47	37	(224,224)	51	51	51	51	51
18	(112,196)	36	36	36	36	37	38	(224,252)	57	55	54	54	53
19	(112,224)	54	52	52	52	51	39	(237,280)	0	0	0	0	0
20	(112,252)	37	37	37	37	37	40	(274,51)	44	45	46	46	47

5.3.3 Resource occupation: Resource utilization is one of the important indicators to measure FPGA systems, such as the flip-flop (FF), look-up-table (LUT) and DSP48s. The resources utilization of the bilinear interpolation, 16-2D-CCI and PP-2D-CCI scheme are analysed, respectively. In the bilinear interpolation method, 27218 LUTs, 45823 registers, 456 RAM/FIFO, and 267 DSP48s are used with utilization rates of 4.45%, 3.74%, 30.40%, and 7.42%,

respectively. There are 69371 LUTs, 107964 registers, 1368 RAM/FIFO, and 481 DSP48s are used with utilization rates of 11.34%, 8.82%, 30.40%, and 7.42%, respectively in the 16-2D-CCI algorithm. In PP-2D-CCI scheme, 52932 LUTs, 84314 registers, 1026 RAM/FIFO, and 292 DSP48s are used with utilization rates of 8.65%, 6.89%, 68.4%, and 10.59%, respectively. The results of the utilization ratio of the logical unit for the bilinear interpolation, 16-2D-CCI and PP-2D-CCI scheme are shown in Figure 15. From Figure 15, the utilization rate of PP-2D-CCI scheme is higher than that of bilinear interpolation algorithm, but lower than that of 16-2D-CCI method. Comparison with PP-2D-CCI and 16-2D-CCI, LUTs, register, memory, and DSP48s are reduced by 16439, 23650, 342, and 89, respectively. The reduction rates are 23.7%, 21.9%, 25% and 18.5%, respectively.

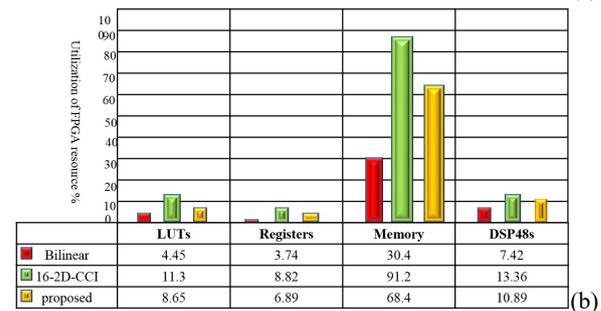
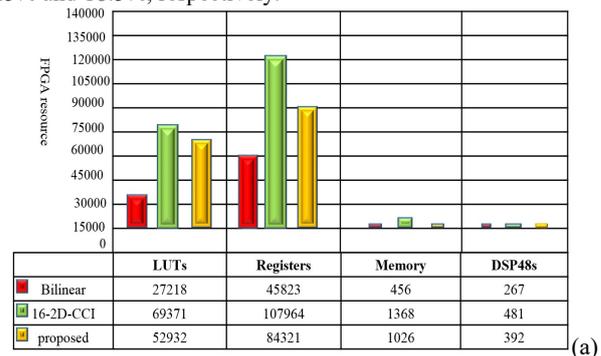


Figure 15. The utilization ratio of logic units for bilinear interpolation method, 16-2D-CCI algorithm, and PP-2D-CCI scheme. (a) The occupied FPGA resources. (b) The utilization ratio of logic units.

5.3.4 Processing speed comparison: The processing speed is one of the most important factors on-board georeferencing [6]. To evaluate the speedup of the proposed method. This paper compares the speedup of the bilinear interpolation, 16-2D-CCI and proposed PP-2D-CCI FPGA-based with PC-based. The clock frequency is defined as 100MHz in FPGA architecture. Table 2 lists the computational time under the different platform. As seen from Table 2, The proposed method can achieve 25 times speedup compared with the PC-based implementation.

Table 2. The computational time

	Bilinear	16-2D-CCI	Proposed
FPGA - Based	0.034	0.063	0.047
PC-Based	1.12	2.05	1.35

6. CONCLUSION

This paper develops an improved to the cubic convolution interpolation based on FPGA. The PP-2D-CCI method is a new technique for resampling of RS image. It has several desirable features than others interpolation scheme.

The PP-2D-CCI method is an improved 16-2D-CCI algorithm, not only improves the speed but also reduces the occupied FPGA resources and the cost. Comparing PP-2D-CCI with 16-2D-CCI resource usage, LUTs, register, memory, and DSP48s were reduced by 16439, 23650, 342, and 89, respectively, the reduction rates are 23.7%, 21.9%, 25% and 18.5%, respectively. The speedup time of the proposed Implementation is about 25 times higher than that when using the PC-based. Therefore, the new scheme has lower computational complexity and higher interpolation speed while keeping the same precision as original CCI.

ACKNOWLEDGEMENTS (OPTIONAL)

This paper is financially the Natural Science Foundation of China under Grant numbers: 41431179, 41601365, 4143379; GuangXi Key Laboratory for Spatial Information and Geomatics Program (Contract No.17-259-16-09); GuangXi innovative Development Grand Grant under the grant numbers: GuiKe AA18118038, GuiKeAA18242048; the National Key Research and Development Program of China under Grant numbers 2016YFB0502501 and the BaGuiScholars program of Guangxi (Guoqing Zhou).

REFERENCES

- Rifman, S.S., 1973. Digital rectification of ERTS multispectral imagery. NASA. Goddard Space Flight Center Symp. On Significant Results obtained from the ERTS-1, Vol. 1, Sect. A and B, 1131-1142, NASA, United States.
- Rifman; S.S. McKinnon, D.M., 1974. Evaluation of digital correction techniques for ERTS images. TRW corporation final Report, TRW 20634-6003-TU-00, NASA Goddard Space Flight Center. Greenbelt, Maryland, March, 1-22.
- Keys Robert, 1981. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6), 1153-1160. doi.org/10.1109/TASSP.1981.1163711.
- Park, S.K., Schowengerdt, R.A., 1982. Image reconstruction by parametric cubic convolution. *Computer Graphics and Image Processing*, 23(3), 258–272. doi.org/10.1016/0734-189X(83)90026-9.
- Zhou, G., Jiang, L., Huang, J., Zhang, R., Liu, D., Zhou, X., Baysal, O., 2018. FPGA-based on-board geometric calibration for linear CCD array sensors. *Sensors*, 18(6), 1794. doi.org/10.3390/s18061794.
- Zhou, G., Zhang, R., Liu, N., Huang, J., Zhou, X., 2017. On-board ortho-rectification for images based on an FPGA. *Remote Sensing*. 9(9), 874. doi.org/10.3390/rs9090874.
- Liu, D., Zhou, G., Huang, J., Zhang, R., Shu, L., Zhou, X., Xin, C.S., 2019. On-Board Georeferencing Using FPGA-Based Optimized Second-Order Polynomial Equation. *Remote Sensing*. 11(2), 124. doi.org/10.3390/rs11020124.
- Zhou, G., 2010. Geo-Referencing of video flow from small low-cost civilian UAV. *IEEE Tran. On Automation Engineering and Science*, 7(1), 156-166, doi.org/10.1109/TASE.2008.2010948.
- Huang, J., Zhou, G., Zhou, X., 2018. A new FPGA architecture of fast and brief algorithm for on-board corner detection and matching. *Sensors*. 18(4), 1014, doi.org/10.3390/s18041014.
- Zhou, G., Yue, T., Shi, Y., Zhang, R., Huang, J. 2016. Second-order polynomial equation-based block adjustment for orthorectification of DISP imagery. *Remote Sensing*, 8(8), 680, doi.org/10.3390/rs8080680.
- Schowengerdt, R.A., 2006. *Remote sensing: models and methods for image processing*. Elsevier,
- Toutin, T., 2004. Geometric processing of remote sensing images: models, algorithms and methods. *International journal of remote sensing*. 25(10), 1893-1924, doi.org/10.1080/0143116031000101611.
- Zhang, K., Whitman, D., 2005. Comparison of three algorithms for filtering airborne lidar data. *Photogrammetric Engineering and Remote Sensing*. 71, March, 313-324; doi.org/10.14358/PERS.71.3.313.
- Arif, F., Akbar, M., 2005. Resampling air borne sensed data using bilinear interpolation algorithm. Proceeding of the 2005 IEEE International Conference on Mechatronics, Taipei, China, 10-12 July, 62-65. doi.org/10.1109/ICMECH.2005.1529228.
- Reichenbach, S.E., Geng, F., 2001. Improved cubic convolution for two dimensional image reconstruction. *2001 IEEE In Nuclear Science Symposium Conference Record*, San Diego, CA, USA, 4-10 November, 1775-1778. doi.org/10.1109/NSSMIC.2001.1008686.
- Pinheiro, M. A., Kybic, J. 2018. Incremental B-spline deformation model for geometric graph matching. In *Biomedical Imaging, 2018 IEEE 15th International Symposium on*, Washington, DC, USA, 4-7 April, 1079-1082. doi.org/10.1109/NSSMIC.2001.1008686.
- Meijering, E., Unser, M., 2003. A note on cubic convolution interpolation. *IEEE Transactions on Image processing*, 12(4), 477-479. doi.org/10.1109/TIP.2003.811493.
- Shi, J. S., Reichenbach E., 2006. Image interpolation by two-dimensional parametric cubic convolution. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 15(7), 1857-1970, doi.org/10.1109/TIP.2006.873429.
- Zhou, D., Shen, X., Dong, W., 2012. Image zooming using directional cubic convolution interpolation. *IET image processing*, 6(6), 627-634, doi.org/10.1049/iet-ipr.2011.0534.
- Hilal, A., 2018. Image re-sampling detection through a novel interpolation kernel. *Forensic science international*, 287, 25-35, doi.org/10.1016/j.forsciint.2018.03.024.
- Zhou, G., Baysal, O., Kaye J., Habib, S., Wang, C., 2004. Concept design of future intelligent earth observing satellites. *International Journal of Remote Sensing*, 25, 2667-2685, doi.org/10.1080/0143116031000101558.
- Nuno-Maganda, M.A., Arias-Estrada, M.O., 2005. Real-time FPGA-based architecture for bicubic interpolation: an application for digital image scaling. *Reconfigurable Computing and FPGAs*, International Conference on

(RECONFIG), Puebla City, Mexico, 28-20 September 2005, 1.
doi.org/10.1109/RECONFIG.2005.34.

Lin, C.C., Sheu, M.H., Chiang, H. K., Liaw, C., Wu, Z. C., 2008. The efficient VLSI design of bi-cubic convolution interpolation for digital image processing. IEEE International Symposium on Circuits and Systems. Seattle, WA, USA, 18-21 May 2008, 480-483. doi.org/ 10.1109/ISCAS.2008.4541459.

Dawood, A.S., Visser, S.J., Williams, J.A., 2002. Reconfigurable FPGAs for real time image processing in space. In 2002 14th International Conference on Digital Signal Processing Proceedings. Santorini, Greece, Greece, 1-3 July, 845-848. doi.org/ 10.1109/ICDSP.2002.1028222.

Zhang, R.; Zhou, G.; Zhang, G.; Zhou, X.; Huang, J., 2018. RPC-Based orthorectification for Satellite Images Using FPGA. *Sensors*, 18, 2511. doi.org/10.3390/s18082511.

Gonzalez, C., Resano, J., Plaza, A.; Mozos, D., 2012. FPGA implementation of abundance estimation for spectral unmixing of hyperspectral data using the image space reconstruction algorithm. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(1), 248-261, doi.org/10.1109/JSTARS.2011.2171673.

Báscones, D., González, C., Mozos, D., 2018. FPGA Implementation of the CCSDS 1.2. 3 Standard for Real-Time Hyperspectral Lossless Compression. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018, 11, 1158-1165. doi.org/10.1109/JSTARS.2017.2767680.

H. Hou; H., 1978. Andrews; Cubic splines for image interpolation and digital filtering. *IEEE Transactions on acoustics, speech, and signal processing*, 26(6), 508-517. doi.org/10.1109/TASSP.1978.1163154.

Shlien, S., 1979. Geometric correction, registration, and resampling of Landsat imagery. *Canadian Journal of Remote Sensing*, 5, 74-89. doi.org/10.1080/07038992.1979.10854986.

Bailey, D. G., 2011: Design for embedded image processing on FPGAs. John Wiley & Sons.

SHANGGUAN J.T., 2015. Improve cubic convolution interpolation algorithm with less points. *Journal of Shanxi University*, 38, 79-84.

Zhang, Y.S., Li, Y.H., Zhen, J.; Li, J.h., Xie, R., 2010. The hardware realization of the bicubic interpolation enlargement algorithm based on FPGA. In Information Processing (ISIP), 2010 Third International Symposium on, IEEE, Qingdao, China, 15-17 October, 277-281. doi.org/10.1109/ISIP.2010.82.

Park, S.K., Schowengerdt, R. A., 1982. Image sampling, reconstruction, and the effect of sample-scene phasing. *Applied Optics*, 21(17), 3142-3151. doi.org/ 10.1364/AO.21.003142.